TITLE OF THE INVENTION

KALMAN TRACKING OF COLOR OBJECTS

BACKGROUND OF THE INVENTION

The present invention relates to the processing of video image sequences, and more particularly to a semi-automatic method for Kalman tracking of color objects within the video image sequence.

With the advent of digital television and the resulting large bandwidth requirements for baseband video signals, compression techniques become ever more important. The currently accepted standard for television compression that provides the most compression while still resulting in acceptable decoded images is the MPEG-2 standard. This standard compresses an image using one of three types of compressed frames — an independently compressed frame, a predictively compressed frame and a bi-directional predictively compressed frame. This standard operates on the images as a whole.

However the content of images may be composed of several objects, such as tennis players and a ball, in front of a background, such as spectators. It is posited that if the objects (tennis players and ball) are separated out from the background (spectators), then the objects may be compressed separately for each frame, but the background only needs to be compressed once since it is relatively static. To this effect many techniques have been proposed for separating objects from the

background, as indicated in the recently published proposed MPEG-7 standard.

Just separating the objects is not sufficient — the objects need to be tracked throughout a given sequence of images that make up a scene. What is desired is a method for tracking objects within a video image sequence.

BRIEF SUMMARY OF THE INVENTION

Accordingly the present invention provides Kalman tracking of color objects within a video image sequence. Objects are separated on the basis of color using a color separator, and a user identifies an object or objects of interest. The object(s) are tracked using a Kalman prediction algorithm to predict the location of the centroid of the object(s) in successive frames, with the location being subsequently measured using a mass density function and then filtered to provide a smooth value for centroid location and velocity. If one of the assumptions for the tracking algorithm fails, then an error recovery scheme is used based upon the assumption that failed, or the user is asked to re-initialize in the current frame.

The objects, advantages and other novel features of the present invention are apparent from the following detailed description when read in conjunction with the appended claims and attached drawing.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

Fig. 1 is a basic block diagram view of an algorithm for Kalman tracking of color objects according to the present invention.

Fig. 2 is an illustrative view for separating objects by color according to the present invention.

Fig. 3 is an illustrative view of the final separation by color according to the present invention.

Fig. 4 is an illustrative view of Kalman prediction of an object centroid from frame to frame according to the present invention.

Fig. 5 is an illustrative view of one type of failure of the tracking algorithm requiring error recovery according to the present invention.

Fig. 6 is an illustrative view of a search pattern for locating the object shown in Fig. 5 according to the present invention.

Fig. 7 is a more detailed block diagram view of the Kalman tracking algorithm according to the present invention.

Fig. 8 is an illustrative view of developing an alpha map for error recovery according to the present invention.

Fig. 9 is an illustrative view of defining the object around the predicted centroid as part of error recovery according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In performing semi-automatic tracking of colored objects in a given

video image sequence, a user indicates in one or more key frames a set of one or more colored objects. The user also indicates other regions of significant size and different colors in the video image sequence. The objects are separated based upon color, and a tracking algorithm then tracks the movements of the indicated objects over time through the video image sequence. This tracking is achieved by associating a Kalman tracking model to each object. The basic algorithm is shown in Fig. 1.

An input video image sequence is input to a color segmentation algorithm, such as that described in co-pending U.S. Patent Application Serial No. 09/270,233 filed March 15, 1999 by Anil Murching et al entitled "Histogram-Based Segmentation of Objects from a Video Signal Via Color Moments". This algorithm uses a hierarchical approach using color moment vectors. The color segmentation algorithm segments the images in the input video image sequence into regions/classes of uniform color properties. Then a Kalman tracking algorithm is applied to each of the segmented objects to produce object "tracks" from one frame to the next of the video image sequence.

As shown in Fig. 2 color segmentation is performed using key rectangles that the user places within different objects of interest, as well as other regions that have significant size and are different in color from the objects. If there are a total of N_u different colors indicated by the user, then the color segmentation algorithm classifies each small block PxQ (P=Q=2 pixels, for example) of each frame of the input video image

sequence into one among the N_u classes or into a "garbage" class. Kalman tracking may be thought of as a post-processing operation on this segmentation result.

~~Kalman tracking applies a~~ Newtonian motion model to the centroids of the objects of interest. As an example, the objective is to track object #K in Fig. 3, whose location in the starting frame I_0 of the input video image sequence is identified by the user. Object #K belongs to color model #A while a different object #L belongs to color model #B. The user "clicks" on the estimated location of the centroid (geometric center) of the object #K. The Kalman state vector at time "n" is:

$$\zeta_k[n] \triangleq \begin{vmatrix} x_k[n] \\ \\ y_k[n] \\ \\ v_{xk}[n] \\ \\ v_{yk}[n] \end{vmatrix}$$

where (x_k, y_k) are the location coordinates of the centroid for object #K, and (v_{xk}, v_{yk}) are the velocity components of object #K. The Newtonian motion model for all objects assumes that acceleration is a white-noise process. This motion model is well known in the art and may be found in the ~~literature on Kalman filtering~~.

With this motion model a state-transition equation becomes:

$$\zeta_k[n+1] = \underline{F}\ \zeta_k[n] + \underline{G}\ \eta_k^s[n]$$

where $\underline{F}$ and $\underline{G}$ are vector constants and $\eta_k^s[n]$ is a stationary, independent, white noise vector with mean: $E\{\eta_k^s[n]\} = \underline{0}$.

A correlation vector bandwidth $\underline{R}_k^s = E\{\underline{\eta}_k^s[n]\underline{\eta}_k^s[m]^T\} = | \sigma_{xk}^2, 0; 0, \sigma_{yk}^2 |$. The noise variances are estimated from the input video sequence.

Through tracking, the position of the centroid of the object #K in the next frame is measured, so:

$$\underline{\Psi}_k[n+1] = \underline{H}\,\zeta_k[n+1] + \underline{\eta}_k^o[n+1]$$

where $\underline{\eta}_k^o[n]$ is the stationary, independent, observation noise vector with means equal to $\underline{0}$, and $\underline{H}$ is a vector constant. Again there is a correlation vector $\underline{R}_k^o$ with noise variances that are estimated.

In steady state tracking the object #K has been tracked to frame I_n and its position and velocity are known. From this point the first step is Kalman prediction. To locate the object #K in frame I_{n+1}

$$(\text{Predicted})\zeta_k'[n+1 \mid n] = \underline{F}\ (\text{filtered})\zeta_k''[n \mid n]$$

The first two entries in $\zeta_k'[n+1 \mid n]$ give the predicted position of the centroid in frame I_{n+1}. Segment PxQ blocks of I_{n+1} into the many colors and identify all the blocks that belong to color model #A — object #K has this color. Then starting from the predicted position, extract a connected set of PxQ blocks that all belong to the color model #A.

The set of connected blocks identified in the first step constitute the desired detection/tracking of the object of interest in frame I_{n+1}. The second step is to measure the centroid position, performed by:

$$_{\Psi}x_k[n+1] = \Sigma x_k Y_k / \Sigma Y_k$$

$$_{\Psi}y_k[n+1] = \Sigma y_k Y_k / \Sigma Y_k$$

where Y is luminance data in frame I_{n+1}. Calculate the centroid position

by using luminance as a "mass density" function. This improves the robustness of the tracking algorithm. Either of the color components may also be used as mass density functions.

Both the measurement and prediction steps are susceptible to noise, so a third step is to filter/smooth the state information. The familiar Kalman filtering equations are used:

$$\zeta_k''[n+1|n] = \zeta_k'[n+1|n] + \Sigma_k[n+1|n]\underline{H}^T(\underline{H}\,\Sigma_k[n+1|n]\underline{H}^T + \underline{R}_k^{o})^{-1} * (\Psi_k[n+1] - \underline{H}\,\zeta_k'[n+1|n])$$

$$\Sigma_k[n+1|n+1] = \Sigma_k[n+1|n] - \Sigma_k[n+1|n]\underline{H}^T(\underline{H}\,\Sigma_k[n+1|n]\underline{H}^T + \underline{R}_k^{o})^{-1} * \underline{H}^T\,\Sigma_k[n+1|n]$$

$$\Sigma_k[n+1|n] = \underline{F}\,\Sigma_k[n|n]\underline{F}^T + \underline{G}\,\underline{R}_k^{s}\underline{G}^T$$

From these equations the filtered/smoothed position and velocity of the centroid of object #K in frame I_{n+1} are obtained. The same process is repeated for each succeeding frame.

~~For the initialization of the process the position of the centroid in~~ frame I_0, $\zeta_k''[0|0]$, is determined. The user "clicks" ~~near~~ the visually estimated geometric center of the object #K, and that point serves as the initial position. The initial velocity is set to zero. Then vales for $\underline{R}_k^s$, $\underline{R}_k^o$ and $\Sigma_k[0|0]$ are determined experimentally and used to determine the centroid position. One such set is

$\underline{R}_k^s = |2.0, 0; 0, 8.0|$; $\underline{R}_k^o = |1, 0; 0, 2|$; $\Sigma_k[0|0] = |1.6, 0, 0, 0; 0, 3.2, 0, 0; 0,$ ~~0, 2.0, 0; 0, 0, 0, 4.0 |.~~

Sub B2

Although the above equations ostensibly give the predicted position of the centroid of object #K in the new frame I_{n+1}, it is possible that these

coordinates lie outside the image field of view. This is easily detected and is an indication to the user that the object of interest has exited the field of view, which is a perceptually significant event. In the algorithm above use the last known "good" position and attempt to delete the object in frame I_{n+1} at that location. If successful, the algorithm continues. Otherwise the algorithm prompts the user to either (a) verify that the object has left the field of view, and hence stop tracking it, or (b) re-initialize at frame I_{n+1} because the tracker model has broken down.

~~Sometimes, due to the geometric shape of the object or due to~~ sudden changes in acceleration, the Kalman prediction points to a centroid location that is outside the boundary of the object #K, as shown in Fig. 5. This situation arises when the PxQ block that contains the predicted centroid position is classified by the color segmentation algorithm as belonging to a class other than color model #A. Again this situation is easily detected. To recover from this, search around a local neighborhood of the predicted centroid position. As shown in Fig. 6, begin at the PxQ block that contains the predicted centroid position and examine PxQ blocks in a spiral search pattern until one is found that belongs to color model #A. Then grow a connected region around this block and label it as object #K in frame I_{n+1}. The radius of the spiral search is a parameter that may be adjusted for each input video image sequence. If the objects of interest move slowly and are "convex" in shape, than a ~~small search radius,~~ such as a 5x5 ~~neighborhood~~, is generally ~~sufficient. If~~

Sub B3

~~there is very rapid and random motion, then a larger search range is desired.~~

The Kalman tracking algorithm is based upon the following assumptions: (I) objects of interest have regular shapes, i.e., cannot track spokes of a bicycle wheel as they are too "thin"; (ii) objects of interest have smooth color, i.e., no stripes or strange patterns; (iii) objects are moving "regularly", i.e., not Brownian motion of gas molecules; and (iv) objects do not occlude each other. When both the out of field of view and outside object boundary error recovery schemes described above fail, then the Kalman tracker is said to have failed. At this point one of the above assumptions has failed. The options at this point are (I) detect all connected regions in frame I_{n+1} that have color model #A, sort according to size/shape and try to locate the desired object #K among them, or (II) ask the user for help, i.e., prompt the user to re-initialize the tracking algorithm at frame I_{n+1}.

Sub B4

~~For option (I) the color segmentor outputs a segmentation map~~ S_{n+1}. Each sample in S_{n+1} represents a spatially corresponding PxQ block of frame I_{n+1}. The value of the sample "n" is {0, 1, . . ., N_u}, where {1, . . ., N_u} are the color models provided to the color segmentor and {0} represents "garbage". The segmentation map is converted to a binary alpha map α_{n+1} by tagging all samples in S_{n+1} that have the same color model as object #K. Thus pixels in α_{n+1} have a value 255 if their corresponding PxQ block in ~~I_{n+1} has the same~~ color as object #K, and have a value of 0 otherwise. ~~The~~

alpha map is fed to a "grow connections" algorithm along with the block coordinates of the predicted position of the centroid of object #K. The output is the desired connected region that is tagged as the object of interest. A simple error recovery scheme begins by detecting all connected regions in frame I_{n+1} that have the same color as object #K, and then selects the biggest one among them.

Thus the present invention provides for Kalman tracking of color objects in an input video image sequence by segmenting the image in the initial frame into a group of objects according to color, determining the position of the centroid of an object of interest and tracking the object through successive frames; and also provides some simple error recovery schemes if the object moves out of the field of view, the predicted centroid falls outside the boundaries of the object or the algorithm breaks down.